محاضرة 9 : spatial filtering

المحتوى : ① Mechanics of filtering

② Linear & non Linear filters

③ Smoothing Spatial filters

Mechanics of spatial filtering

* أولا: أفرق بين ال filtering في ال frequency Domain وال Spatial Domain
* ال frequency filter بيبقى (Low, High, Band) pass أو Band reject (هنشرحها المحاضرة 11)
* ال Spatial filters منها نوعين:
  - Linear filter → linear operation
  - non-Linear filter → non-Linear operation
* ال Spatial filtering بيشتغل على كل ال pixels واحد ورا التاني، بيبقى ال filter عندي له أسماء زي Mask, Kernel, Window
* ال window أبعادها لازم تكون رقم odd، [m x n] ال n وال m أرقام فردية فبنقول $\boxed{n=2b+1 \quad , \quad m=2a+1}$ بحيث مهما غيرت في a, b هتدي دايما نتائج فردية للمعادلات
* هنرمز لل window ب $w(s,t)$ والصورة الأصلية $f(x,y)$ والصورة بعد الـ filter بـ $g(x,y)$
* لو فرضت عندي window أبعادها 5 x 5 (عادة بنختارها مربعة)

نقول $m=5$, $n=5$، ودايما $-a \le s \le +a$ و $-b \le t \le b$

ـ في حالتنا دلوقت ال $a = 2$ وال $b = 2$ من معادلاتهم فوق

n

m

w(-a,-b)	w(-a,-1)	w(-a,0)	w(-a,1)	w(-a,b)
w(-1,-b)	w(-1,-1)	w(-1,0)	w(-1,1)	w(-1,b)
w(0,-b)	w(0,-1)	w(0,0)	w(0,1)	w(0,b)
w(1,-b)	w(1,-1)	w(1,0)	w(1,1)	w(1,b)
w(a,-b)	w(a,-1)	w(a,0)	w(a,1)	w(a,b)

$-a = -2$
$-b = -2$

لاحظ إنك في المصفوفة ال $s = t = 0$ وال s بتغيرها من $-a \to a$ بتحركه في اتجاه x، ونفس الكلام مع t في اتجاه y بقيم من $-b \to b$

* لص كويس بقى على شكل ال filter في سلايد [4 . 9] وراجعها

* لما بطبق ال filter بطبقه على f(x, y) ، بس لازم آخد ال neighbors بتوعها على قد ال window

* لما نيجي نطبق ال filter هنضرب كل ما في f المناظرة ليها ونجمع

$$
\begin{aligned}
g(x,y) = \; & f(x-1, y-1) * w(-1,-1) \\
& + f(x-1, y) * w(-1, 0) \\
& + f(x-1, y+1) * w(-1, 1) \\
& + f(x, y-1) * w(0, -1) \\
& + f(x, y) * w(0, 0) \\
& + f(x, y+1) * w(0, 1) \\
& + f(x+1, y-1) * w(1, -1) \\
& + f(x+1, y) * w(1, 0) \\
& + f(x+1, y+1) * w(1, 1)
\end{aligned}
$$

So, we can write g(x, y) as

$$
g(x, y = \sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x+s, y+t)\, w(s,t)
$$

$$
= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)
$$

بغير ال x, y عشان ال window تلف على كل البكسلز في الصورة
ولو حصل إني كنت عند الأطراف وفيه بكسلز مهلهاش قيمة ، بافرضها بـ صفر مثلا

* الطريقة اللي فوق دي بنسميها sum of products وهي طريقة Linear

* ممكن نعمل الكلام اللي فوق بطريقتين → Convolution ، Correlation

* المعادلة اللي فاتت هي ال Correlation

$$w(x,y) \text{☆} f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

* عملية ال Convolution هي نفسها عملية ال Correlation بس بلف ال window بـ 180°

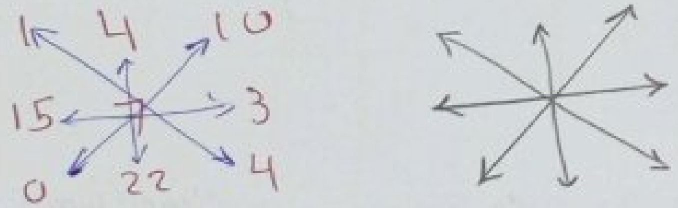
$$w(x,y) \bigstar f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x-s, y-t)$$

يعني لو عندي شكل window كالتالي (نشوف الفرق)

Correlation window $\xleftrightarrow{\text{rotate } 180^\circ}$ Convolution window

$$\begin{matrix} 1 & 4 & 10 \\ 15 & 7 & 3 \\ 0 & 22 & 4 \end{matrix} \qquad \begin{matrix} 4 & 22 & 0 \\ 3 & 7 & 15 \\ 10 & 4 & 1 \end{matrix}$$

عشان تسهل على نفسك ال rotation اعمل عملية تبديل لل pixels بالشكل ده



$$\begin{matrix} 4 & 22 & 0 \\ 3 & 7 & 15 \\ 10 & 4 & 1 \end{matrix}$$

رياضيا ، بنعمل عملية ال rotation بالضرب بال unit impulse (يعني قيمة البكسل هي 1)
وسلايد [9.6] بتوضح الكلام ده على vector ، بينما سلايد [9.7]
بتوضح الكلام ده على صورة عادية

* بتقول إنه عملية ال Convolution وال Correlation عمليات
Commutative ، يعني ترتيبهم مش فارق

$$w(x,y) \text{☆} f(x,y) = f(x,y) \text{☆} w(x,y) \quad \text{(Correlation)}$$

$$w(x,y) \bigstar f(x,y) = f(x,y) \bigstar w(x,y) \quad \text{(Convolution)}$$

* ممكن نكتب شكل ال filter بـ Matrix Notation

$$\underbrace{\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_{mn} \end{bmatrix}}_{w(x,y)} * \underbrace{\begin{bmatrix} z_1 & z_2 & z_3 & \vdots & \vdots \\ z_4 & z_5 & z_6 & \vdots & \vdots \\ z_7 & z_8 & z_{mn} & \vdots & \vdots \\ - & - & - & \vdots & \vdots \end{bmatrix}}_{f(x,y)}$$

لو هطبق الفلتر w على $f(x,y)$ وعندي في ال $f(x,y)$ مجموعة ال (z) هي اللي داخلة في عملية الحساب (يعني هي اللي هطبق عليها ال window)

ممكن أحول العملية لـ vector طوله $mn \times 1$

$$w = [w_1 \quad w_2 \quad w_3 \quad w_4 \quad \cdots \quad w_{mn}]$$

$$z = [z_1 \quad z_2 \quad z_3 \quad z_4 \quad \cdots \quad z_{mn}]$$

وأعمل ضرب Matrix بالشكل التالي

البكسل اللي هتتغير قيمته

$$\boxed{R = w^T z}$$

$$= \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{mn} \end{bmatrix} [z_1 \quad z_2 \cdots z_{mn}]$$

$$= w_1 z_1 + w_2 z_2 + w_3 z_3 + \cdots + w_{mn} z_{mn}$$

* خلي عندك مرونة في المسميات في المعادلات ، الفكرة ثابتة ، وتتغير في الرموز كتير عادي جداً.

---

* إزاي نعمل ال filter أو ال Mask بتاعي؟
* شوف سلايد [9.10] ، فيه التفاصيل وموديك نوعين من الـ filters
* على إشتمال averaging filter خدناه مرة ، كل اللي هيعمله إنه هيضرب قيمة البكسل في 1 وهيجمعهم ويضربهم في $\frac{1}{9}$ (أو يقسم على مجموع ال weights بتاعت ال window اللي هي مجموع الواحايد بـ 9)

* على المجسم ، هسميه weighted average ، حسب ان ال weighted filter دا انه مدي weight للبكسلز ، اللي في النص مديله weight بـ 4 ، واللي في الزوايا ليهم weight=1 والباقي weight=2

* هنضرب كل بكسل في ال weight ونجمع ونقسم على مجموع الـ weights

* لاحظ ان ال weight بـ 4 لل center بيخلي تأثير ال pixel اللي في الـ center هو الأكبر وهكذا ...

* هنشوف مثال عملي لتطبيق الـ filters على صورة لما نوصل لل smoothing

[المثال مهم لأن الدكتور عمله في المحاضرة مسألة للطلبة]

* ممكن الأرقام بتاعت ال window [هسميها ال coefficients] بدل ما نجيبها بإيدينا ، نحسبها من معادلة ، زي في ال Gaussian filter = Gaussian blur

هنسمي ال filter coefficient بـ $h(x,y)$ أو خليها $h(s,t)$ عشان متتلخبطش ، بحسبها واحدة لل filter واستخدمها على الصورة براحتي

h(-1, -1)	h(-1, 0)	h(-1, 1)
h(0, -1)	h(0, 0)	h(0, 1)
h(1, -1)	h(1, 0)	h(1, 1)

$$h(s,t) = e^{\frac{-(s^2+t^2)}{2\sigma^2}}$$

حيث σ هي ال standard deviation
وال σ^2 هي ال Variance

* عادة بيدي ال σ في الامتحان ، هنفرضها بـ 2 ونحسب الـ Coefficients

$$h(-1,-1) = e^{\frac{-((-1)^2+(-1)^2)}{2(2)^2}} = e^{\frac{-2}{8}} = e^{-1/4} = 0.77$$

simlarly calculate the remaining $h(s,t)$

متنساش انك لما تطبق الفلتر تقسم على مجموع ال Coeffecients عشان تحافظ على قيم البكسل متعداش المسموح (L - 1)

Always divide by $\left[\sum_{s=-a}^{a} \sum_{t=-b}^{b} h(s,t)\right]$

Linear vs non Linear filters

* بشكل عام، عملية ال filtering سواء كانت linear أو لا فمفيش فرق كبير من حيث إنه فيه window وبيشتغل على كل ال pixels.
* الفرق بيكون في ال operation اللي بعملها

* ال Standard Operation لعمليات ال Linear filter بتكون [ركز كويس]
- Sum of products على ال filter Coeffeciants
- عادة بتكون ال correlation
- ممكن نستخدم ال Convolution [بلف ال window ب 180°]

* ممكن نبقى العملية non-Linear filtering زي ال median [القيمة المتوسطة، بترتب الأرقام تصاعدي أو تنازلي وآخد القيمة اللي في النص، ولو العدد زوجي بتاع الأرقام باخد الرقمين في النص وأجمعهم وأقسم على 2] أو إيجاد ال Maximum Intensity

---

Smoothing spatial filters

* عملية ال Smoothing هي نفسها ال blurring هي نفسها noise reduction هي نفسها low pass filtering
* عادة بنستخدمها في ال noise reduction زي ما عملنا بال averaging قبل كده لما قولنا إنه تطبيق على عملية ال Addition

[ذاكر محاضرة 9 من ال slides وصفحات 39, 38, 37 لما تخلص الورق ده، ضروري]

* سلايد 9.16 فيها تطبيق ال filter ، كلما يكبر أبعاد ال window بيعمل blurring أكتر عشان بياخد pixels أكتر في الحساب

* سلايد [9.17] عمل فيها Smoothing وبعدها Thresholding

* سلايد [9.18] عمل فيها noise reduction باستخدام ال median filter عشان يشيل Salt and pepper noise [اعرف ال noise دي كويس أوي]

* ممكن نستخدم ال median filter عشان يشيل ال speckle noise برده
(معرفش الدكتور جاب سيرتها ولا لأ)

* ال Salt & pepper noise بتبقى عبارة عن pixels كتيرة بتخلي الصورة كأنها مشوشرة زي التلفزيونات القديمة كده (:D) وسلايد 9.18 بتوضح الفرق

---

* لو رجعنا لسلايد 9.15 هنلاقي فيها صيغتين للفلتر، واحدة normalized والتانية لأ، ونفضل ال unnormalized عشان أسرع في الحساب

window

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

normalized

بقسم ال Coeffecients على وأعدل ال window وبعدين أحسب العملية

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

window

1	2	1
2	4	2
1	2	1

unnormalized

$\frac{1}{16}$ اقسم عليه لما تحسب

بطبق ال window ولما أخلص الحساب بقسم على مجموع ال coeffecients مرة واحدة

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

مجموع ال coeffecients $= 16$

هنطبق الـ filters الثلاثة (average, median, gaussian)

على pixel في صورة ومع فرض σ ب 0.5 لل gaussian والـ Mask 3X3

5	7	3	4
6	7	8	10
12	6	10	2

* ملاحظة مهمة ، لو وقت هنطبق الـ filter ، وهنغير قيمة الـ pixel في النص (7) لحاجة جديدة ، لما أحرك الـ window عشان أشتغل على البكسل $\boxed{8}$ هاخد القيمة القديمة (7) للفلتر ، مش القيمة الجديدة بعد الـ filter ، فخلي بالك أوي صدي عشان متغلطش في الامتحان

① average

هنوضح شكل الـ Mask والمعادلات لو محتاجة ، وهنشتغل unnormalized

w(-1,-1)

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \xrightarrow{\text{ضرب}} \begin{bmatrix}5&7&3\\6&7&8\\12&6&10\end{bmatrix}$$

$a = 1$

$b = 1$

$m = 2a+1 = 3$

$n = 2a+1 = 3$

$f(x,y) = 7$

required $g(x,y)$

* هنضرب كل pixel في الـ window في اللي قصاده من الصورة

المعادلة هتبقى

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

we have $w(-1,-1) = w(-1,0) = \cdots\ w(1,1) = 1$

$$g(x,y) = \frac{(1)(5)+(1)(7)+(1)(3)+\cdots+(1)(10)}{1+1+1+1+1+1+1+1+1} = \frac{64}{9} = 7.11 \simeq 7$$

② gaussian

0.0813	0.1353	0.0813
0.1353	1	0.1353
0.0813	0.1353	0.0813

$*$

5	7	3
6	7	8
12	6	10

$f(x,y)$

$\boxed{g(x,y) = 0}$

(*) $w(s,t) = e^{-\frac{(s^2+t^2)}{2\sigma^2}}$

* نفس الحاجة فوق بس هنغير الـ coefficient ونحسبها من المعادلة (*)

$$g(x,y) = \frac{0.0813(5+3+12+10)+0.1353(7+6+8+6)+7}{1+0.1353*4+0.0813*4}$$

$w(-1,-1) = 0.0183 = 0.004 \to 0$ والباقي زيها

③ median

5	7	3
6	7	8
12	6	10

الجزء اللي هنعدله

$\xrightarrow{\text{ترتيب}}$ تصاعدي أو تنازلي

3	5	6
6	7	7
8	10	12

ناخد القيمة في الـ center بعد الترتيب

هنقول الـ g(x,y) هي القيمة الـ median لـ intensities الموجودة في الـ window

$$g(x,y) = 7$$

مفيش w(s,t) زي المعادلات فوق ومش عارف نكتب معادلة للـ median إزاي هو مش linear عموما فمش هينفع نطبق المعادلات فوق